APPLICATION FOR UNITED STATES PATENT

INVENTORS: Donald C. Englin
Donald W. Mackenthun
Kelvin S. Vartti

INVENTION: METHOD FOR AVOIDING DELAYS DURING SNOOP
REQUESTS

**DOCKET
NUMBER:**   RA 5221 (33012/290/101))

Unisys Corporation
Charles A. Johnson
P O Box 64942 - MS 4773
St. Paul, MN 55164
Attorney for Applicant
Reg. No.: 20,852

SPECIFICATION

METHOD FOR AVOIDING DELAYS DURING SNOOP REQUESTS

CROSS REFERENCE TO CO-PENDING APPLICATIONS

The present application is related to co-pending U.S. Patent Application Serial No. _____, filed _____, entitled Cooperative Hardware and Microcode Control System for Pipelined Instruction Execution; U.S. Patent Application Serial No. _____, filed _____, entitled Method for Improved First Level Cache Coherency; U.S. Patent Application Serial No. _____, filed _____, entitled Leaky Cache Mechanism; and U.S. Patent Application Serial No. _____, filed _____, entitled Data Coherency Protocol for Multi-level Cached High Performance Multiprocessor System, assigned to the assignee of the present invention and incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. **Field of the Invention:** - The present invention relates generally to data processing systems employing multiple instruction processors and more particularly relates to multiprocessor data processing systems employing multiple levels of cache memory.

2. **Description of the Prior Art:** - It is known in the art that the use of multiple instruction processors operating out of

1

common memory can produce problems associated with the processing of obsolete memory data by a first processor after that memory data has been updated by a second processor. The first attempts at solving this problem tended to use logic to lock processors out of memory spaces being updated. Though this is appropriate for rudimentary applications, as systems become more complex, the additional hardware and/or operating time required for the setting and releasing of locks can not be justified, except for security purposes. Furthermore, reliance on such locks directly prohibits certain types of applications such as parallel processing.

The use of hierarchical memory systems tends to further compound the problem of data obsolescence. U.S. Patent No. 4,056,844 issued to Izumi shows a rather early approach to a solution. The system of Izumi utilizes a buffer memory dedicated to each of the processors in the system. Each processor accesses a buffer address array to determine if a particular data element is present in its buffer memory. An additional bit is added to the buffer address array to indicate invalidity of the corresponding data stored in the buffer memory. A set invalidity bit indicates that the main storage has been altered at that location since loading of the buffer memory. The validity bits are set in accordance with the memory store cycle of each processor.

U.S. Patent No. 4,349,871 issued to Lary describes a bussed architecture having multiple processing elements, each having a dedicated cache memory. According to the Lary design, each

2

processing unit manages its own cache by monitoring the memory bus. Any invalidation of locally stored data is tagged to prevent use of obsolete data. The overhead associated with this approach is partially mitigated by the use of special purpose hardware and through interleaving the validity determination with memory accesses within the pipeline. Interleaving of invalidity determination is also employed in U.S. Patent No. 4,525,777 issued to Webster et al.

Similar bussed approaches are shown in U.S. Patent No. 4,843,542 issued to Dashiell et al, and in U.S. Patent No. 4,755,930 issued to Wilson, Jr. et al. In employing each of these techniques, the individual processor has primary responsibility for monitoring the memory bus to maintain currency of its own cache data. U.S. Patent No. 4,860,192 issued to Sachs et al, also employs a bussed architecture but partitions the local cache memory into instruction and operand modules.

U.S. Patent No. 5,025,365 issued to Mathur et al, provides a much enhanced architecture for the basic bussed approach. In Mathur et al, as with the other bussed systems, each processing element has a dedicated cache resource. Similarly, the cache resource is responsible for monitoring the system bus for any collateral memory accesses which would invalidate local data. Mathur et al, provide a special snooping protocol which improves system throughput by updating local directories at times not necessarily coincident with cache accesses. Coherency is assured

3

by the timing and protocol of the bus in conjunction with timing of the operation of the processing element.

An approach to the design of an integrated cache chip is shown in U.S. Patent No. 5,025,366 issued to Baror. This device provides the cache memory and the control circuitry in a single package. The technique lends itself primarily to bussed architectures. U.S. Patent No. 4,794,521 issued to Ziegler et al, shows a similar approach on a larger scale. The Ziegler et al, design permits an individual cache to interleave requests from multiple processors. This design resolves the data obsolescence issue by not dedicating cache memory to individual processors. Unfortunately, this provides a performance penalty in many applications because it tends to produce queuing of requests at a given cache module.

The use of a hierarchical memory system in a multiprocessor environment is also shown in U.S. Patent No. 4,442,487 issued to Fletcher et al. In this approach, each processor has dedicated and shared caches at both the L1 or level closest to the processor and at the L2 or intermediate level. Memory is managed by permitting more than one processor to operate upon a single data block only when that data block is placed in shared cache. Data blocks in dedicated or private cache are essentially locked out until placed within a shared memory element. System level memory management is accomplished by a storage control element through which all requests to shared main memory (i.e. L3 level) are routed. An apparent improvement to this approach is shown in U.S. Patent No.

4

4,807,110 issued to Pomerene et al. This improvement provides prefetching of data through the use of a shadow directory.

A further improvement to Fletcher et al, is seen in U.S. Patent No. 5,023,776 issued to Gregor. In this system, performance can be enhanced through the use of store around L1 caches used along with special write buffers at the L2 intermediate level. This approach appears to require substantial additional hardware and entails yet more functions for the system storage controller.

The special problem with multiple processor, multiple level, high speed, cached memory systems involving coherency may be resolved through the use of SNOOPing. In this process, the control logic for one cache memory interrogates the tag memory corresponding to other cache memories to determine whether a selected data element has been modified by operation of a different processor. Though this procedure is quite useful, the primary difficulty concerns efficiency.

SUMMARY OF THE INVENTION

The present invention overcomes the problems found in the prior art by providing a method and apparatus for improving the efficiency of maintaining coherency within a system having multiple levels of cache memory. This enhancement to efficiency is accomplished by utilizing improving the efficiency of the SNOOP activity.

The preferred mode of the present invention includes up to four main memory storage units. Each is coupled directly to each of up to four "pod"s. Each pod contains a level three cache memory coupled to each of the main memory storage units. Each pod may also accommodate up to two input/output modules.

Each pod may contain up to two sub-pods, wherein each sub-pod may contain up to two instruction processors. Each instruction processor has two separate level one cache memories (one for instructions and one for operands) coupled through a dedicated system controller, having a second level cache memory, to the level three cache memory of the pod.

Unlike many prior art systems, both level one and level two cache memories are dedicated to an instruction processor within the preferred mode of the present invention. The level one cache memories are of two types. Each instruction processor has an instruction cache memory and an operand cache memory. The

6

instruction cache memory is a read-only cache memory primarily having sequential access. The level one operand cache memory has read/write capability. In the read mode, it functions much as the level one operand cache memory. In the write mode, it is a semi-

5   store-in cache memory, because the level two cache memory is also dedicated to the instruction processor.

In accordance with the preferred mode of the present invention, each dedicated system controller has a level two cache memory, a level two cache tag memory, and two (i.e., one for

10 instruction and one for operand) level one cache duplicate tag memories. Unlike past designs, interfaces to the tag memories are established which are separate and independent of the interface to the level two cache memory. Though, for normal accesses, it is intuitive that the level two cache memory and the tag memories are

15 referenced during the same operation, SNOOP operations access only the tag memories. Thus, the present invention provides the opportunity for regulating memory accesses and SNOOP accesses independently.

The present invention gives the SNOOP accesses priority over

20 other uses of the tag memories. Again, this might seem counterintuitive because it tends to slow access requests from the dedicated processor. However, overall system throughput is increased, because SNOOP requests are honored more quickly permitting the other system processors to operate with less SNOOP

25 request latency.

7

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects of the present invention and many of the attendant advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, in which like reference numerals designate like parts throughout the figures thereof and wherein:

FIG. 1 is an overall block diagram of a fully populated system in accordance with the present invention;

FIG. 2 is a schematic block diagram of one pod;

FIG. 3 is a schematic block diagram of one instruction processor along with its dedicated system controller;

FIG. 4 is a detailed diagram showing data flow of the operation of the present invention; and

FIG. 5 is a timing diagram of an example in accordance with the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. 1 is an overall block diagram of fully populated data

5   processing system according to the preferred mode of the present

invention. This corresponds to the architecture of a commercial

system of Unisys Corporation termed "Voyager".

The main memory of the system consists of up to four memory

storage units, MSU 10, MSU 12, MSU 14, and MSU 16. Being fully

10   modular, each of these four memory storage units is "stand-alone"

and independent of one another. Each has a separate point-to-point

dedicated bi-directional interface with up to four "pods", POD 18,

POD 20, POD 22, POD 24. Again, each of the up to four pods is

separate and independent of one another.

15   The contents of POD 20 are shown by way of example. For the

fully populated system, POD 18, POD 22, and POD 24 are identical to

POD 20. The interface between POD 20 and each of the four memory

storage units (i.e., MSU 10, MSU 12, MSU 14, and MSU 16), is via a

third level cache memory designated cached interface, CI 26, in

20   this view. CI 26 couples with two input/output controllers, I/O

Module 44 and I/O Module 46, and two sub-pods, SUB 28 and SUB 30.

A more detailed explanation of the POD 20 is provided below.

The above described components are the major data handling

elements of the system. In the fully populated system shown, there

9

are sufficient components of each type, such that no single
hardware failure will render the complete system inoperative. The
software employed within the preferred mode of the present system
utilizes these multiple components to provide enhanced reliability
5   for long term operation.

The remaining system components are utilitarian rather than
data handling. System Oscillator 32 is the primary system time and
clocking standard. Management System 34 controls system testing,
maintenance, and configuration. Power Controller 36 provides the
10   required electrical power. System Oscillator 38, Management System
40, and Power Controller 42 provide completely redundant backup
capability.

FIG. 2 is a more detailed block diagram of POD 20. The level three cache memory interfaces directly with the memory storage units via TLC Controller 26 (see also Fig. 1). The actual storage for the level three cache memory is TLC SRAMS 48. As indicated this static random access memory consists of eight 16 byte memory chips.

Subpod 28 and subpod 30 each contain up to two individual instruction processors. These are designated Voyager IP 50, Voyager IP 52, Voyager IP 54, and Voyager IP 56. As explained in detail below, each contains its own system controller. In accordance with the preferred mode of the present invention, these instruction processors need not all contain an identical software architecture.

FIG. 3 is a more detailed block diagram of Voyager IP 50, located within Subpod 28, located within POD 20 (see also Figs. 1 and 2). As explained above, each instruction processor has a dedicated system controller having a dedicated level two cache memory. Instruction processor 64 has two dedicated level one cache memories (not shown in this view). One level one cache memory is a read-only memory for program instruction storage. Instruction processor 64 executes its instructions from this level one cache memory. The other level one cache memory (also not shown in this view) is a read/write memory for operand storage.

Instruction processor 64 is coupled via its two level one cache memories and dedicated system controller 58 to the remainder of the system. System controller 58 contains input logic 74 to interface with instruction processor 64. In addition, data path logic 70 controls movement of the data through system controller 58. The utilitarian functions are provided by Locks, Dayclocks, and UPI 62.

The remaining elements of system controller 58 provide the level two cache memory functions. SLC data ram 66 is the data actual storage facility. Control logic 70 provides the cache management function. SLC tags 72 are the tags associated with the level two cache memory. FLC-IC Dup. Tags 76 provides the duplicate tags for the level one instruction cache memory of instruction

12

processor 64. Similarly, FLC-OC Dup. Tags 78 provides the duplicate tags for the level one operand cache memory of instruction processor 64. For a more complete discusses of this duplicate tag approach, reference may be made with the above identified co-pending and incorporated U.S. Patent Applications.

FIG. 4 is a detailed diagram showing data flow of the operation of the present invention. Shown are the significant components of system controller 50 involved in practicing the present invention.

Bus interface logic 60 couples system controller 50 with third level cache controller 26 and the remaining system controllers of Pod 20 (see also Figs. 1 and 2) via memory bus 80. Unlike prior art system controller, bus interface logic 60 provides separate and independent paths to level two storage, SLC Data 66, and the three rams storing tag information, FLC-IC 76, FLC-OC 78, and SLC Tags 72. Internal bus 84 couples to the data storage and internal bus 82 couples to the tag memories. In view of these separate and independent paths, the tag memories (i.e., FLC-IC 76, FLC-OC 78, and SLC Tag 72) may be addressed directly from memory bus 80, without access to the cache storage, SLC Data 66.

Given this independent access, control logic 70 has the ability to direct accesses to the tag memories, the cache storage, or both. In accordance with the present invention, control logic 70 always provides higher priority to SNOOP access requests from memory bus 80 to the tag memories than access requests to the cache storage, which also requires access to the tag memories.

Fig. 5 is a timing diagram of an example of the operation of the present invention. In the preferred embodiment, the instruction processor timing clock cycles are divided into sub-

5   cycles, called slots. In the present view, time proceeds from left to write in the timing diagram. Each vertical column represents one slot. The two horizontal rows associated with Tag slots 94 and Data slots 96 represent the prior art solution in response to data and SNOOP requests, whereas the two horizontal rows associated with

10   Tag slots 98 and Data slots 100 represent the savings associated with the present invention.

In the prior art solution, a first data request is made (i.e., Req. 1) which occupies both the tag slot and data slot for the first time slot. Similarly, the second request occupies both the

15   tag slot and data slot for the second time slot. The request 1 and request two updates occupy the third and fourth time slots. This happens even though a SNOOP request is present. Thus, the SNOOP request is acted upon only during the ninth tag time slot, because there is no new data request present. This delay of the SNOOP

20   request means that other processors within the system remain idle waiting for the response to the SNOOP request, even though the local processor has all of its requests honored on a high priority basis.

According to the present invention, however, as shown in tag

15

slots 98 and data slots 100, the SNOOP request, is honored upon

being presented, because of its high priority. This delays the

responses to the data requests of the local processor. However,

the remote processor do not need to wait unduly for the honoring of

5   the SNOOP request. Thus, overall system throughput is increased.

Having thus described the preferred embodiments in sufficient detail for those of skill in the art to make and use the present invention, those of skill in the art will be readily able to apply

5   the teachings found herein to yet other embodiments within the scope of the claims hereto attached.

WE CLAIM: